14. Information Presentation in Web Applications

14.1 Text Information

14.1.1 Font Style, Size, and Color

The application uses regular font style for text in the page body, and bold style for text in headings and to focus user attention on important information. If bold style is used in the page body, it is indicated by the appropriate font tag (e.g.,) and not with a heading tag. The application does not underline text (since underlining identifies a hyperlink) or use italics in entire sentences or blocks of text (since this style in some fonts can be difficult to read when displayed by a browser).

If the application uses variations in font size, it does so only for emphasis or to focus attention on specific portions of a page and ensures that the text is rendered correctly when displayed by different browsers. If the application uses the <font size> tag, it specifies text in terms of relative rather than absolute font size. The application does not use this tag to change the size of all text on a page since users may set font size in the browser and the application value could conflict with this setting (e.g., reduce legibility by producing an unacceptable size change).

The application can use attributes within the <body> tag to set the color of the page background, some or all of page text, visited and unvisited links, and links when users click on them. Text color can also be set using the <font color> tag. Some browsers allow users to select font colors as a preference setting, with the option for these choices to override the values defined by the application. If the application requires control over the color(s) in which page attributes are displayed, it ensures that the color values it sets are rendered correctly by different browsers and do not conflict with any color settings selected by users (e.g., do not result in unreadable text).

14.1.2 Grammar and Wording

Text in the page body is presented in mixed case, following standard rules for capitalization and punctuation and the use of acronyms and abbreviations, as indicated in sections 12.1.2 and 12.1.3. Dates (e.g., in the page footer) are expressed in a format that is unambiguous internationally (see section 15).

The content of each page is written so that it can stand alone if viewed independently of other pages in the application. For example, a page does not start with references to “the previous step” or “the next step.” Time references (e.g., “the latest version of this document”) are worded so that they do not become outdated. Each page includes opening information (e.g., in the first paragraph) that provides sufficient context for users to understand the content of the page, or the page includes a link to background information that provides this context.

Text is worded so that it is readable (and can be printed, if desired) despite the presence of links on the page. The text does not contain references to online aspects of the content or to protocols or mechanisms used to find the information. For example, “Click here for information on XXX” is better stated as “See the section on XXX.” Similarly, “You can read more about XXX in the

tutorial that is linked to the home page" is better stated as "The tutorial contains more information on this topic." The text used in a link describes what it is about and does not consist solely of the word "here." Because a browser may include the text as a bookmark or hitlist entry, using "here" as a link results in entries that contain only this word.

14.1.3 Headings, Alignment, and Spacing

HTML provides a set of tags for displaying different levels of headings on a page. The application uses the heading levels in order, starting with <h1> for the page name. The same heading tag is applied to the same level of content throughout the application. Text in <h1> and <h2> headings is displayed in upper case or mixed case, and heading text below <h2> in mixed case only. Each heading is worded so that it is meaningful when it is viewed either in isolation (e.g., when users navigate to the page from outside the application) or as an item in a menu or hitlist. The wording uniquely identifies the content that follows the heading so that search engines that examine the keywords in heading text can determine if the page is a "hit" that matches the search parameters.

The default page layout is to center the page name, left-align the headings and text, and left-align or center the graphics; the application can select alternative layout(s) as appropriate to the content being presented. If text is placed adjacent to a graphic, the graphic is placed at the left and the text to its right. Whenever possible, paragraphs are kept short (i.e., four or fewer sentences) to improve readability.

White space is provided among the various elements on the page, with paragraph and line break tags (i.e., <p> and
) used to separate text and graphics, and separators (i.e., <hr>) used to delimit different areas of the page (e.g., the page body and footer). Separators are created as a tag, rather than as an inline image so that they can be displayed by all types of browsers. The default appearance of a separator is a shaded engraved line that extends the width of the page.

14.1.4 Lists

When text information is structured as an ordered or unordered list, HTML precedes each list item with a number or bullet. The application does not use the tags for ordered and unordered lists (i.e., <ol> and <ul>) if the text in a list is already numbered, since these tags add either bullets or numbers to the text that follows.

The application can display a bulleted list either with the <ul> tag or by creating customized bullet graphics. If the latter approach is used to define a set of links (e.g., on a navigation page), the graphic is included as part of the link (so that users can select the graphic or the text, as is the case with bulleted items displayed with the <ul> tag). Using tags is preferred because bullets created in HTML are viewable on all types of browsers, while bullets created as graphics cannot be displayed by text-only browsers or browsers with graphics deactivated.

14.1.5 Tables

A table is placed in the body of a page, immediately following the paragraph in which the table is first referenced, rather than on a separate page. The table includes a title that is left-aligned or

centered above the table. The information in the table is arranged in sequential, spatial, alphabetic, functional, or chronological order. Information that is particularly important or is used frequently is presented first. If the application uses the <table> tag to present critical information, it provides an option for users to view the information as preformatted text so that browsers that do not support the <table> tag can display the information.

Each column in the table has a heading and is clearly separated from information in other columns. If the application uses the <table> tag, it specifies values for "cell spacing" (i.e., the amount of space inserted between cells) and "cell padding" (i.e., the amount of space between the border of the cell and its content). A value of 5 for both attributes is recommended so that table contents are easy to read. A table can be displayed with or without borders (i.e., the lines surrounding each cell in the table). If borders are used, line thickness is minimized (e.g., set to 1 pixel) so as not to reduce the legibility of information in the table. If a table is used to format the text or graphics on a page, cell borders are set to zero so that the table is hidden. Because browsers vary in the alignment defaults for table headings, table cells, and captions, the application should specify values for the "align" and "valign" attributes (rather than relying on the defaults) to ensure that table contents are displayed properly by different browsers.

14.2 Images, Graphics, and Multimedia

14.2.1 Inline and External Images

If the application includes pictures, diagrams, or illustrations on a page, they are presented as inline rather than external images (i.e., as part of the page itself rather than on a separate page). Each image has a transparent background so that it is the same color as the underlying page. Whenever possible, the image is sized so that it can be viewed without scrolling. A short text description is included with the image (e.g., using the "alt" attribute in the <img src> tag) so that users with all-text browsers or browsers with graphics deactivated are informed about what cannot be displayed.

The application specifies the "height" and "width" attributes in the <img> tag to speed up the display of the image and allow the browser to set aside space for the image as the image is downloaded. If the height and width values are incorrect, the image will be distorted when displayed. If desired, the application can specify "hspace" and "vspace" in the <img> tag to insert horizontal and vertical space around an image (so that it does not abut the text adjacent to it). Smaller values of "hspace" add space between the image and text without affecting the position of the image, while larger values insert the space and move the image to the right (i.e., away from the left margin).

If an image is large (either in dimensions or file size), it is accessed as an external image, and the parent page includes a miniature inline image that links to it and provides a "preview" of what it contains. The size of the external image is indicated on the parent page so that users can decide if they want to wait while the image is transferred. If the external image requires a helper application for downloading and display, this information is provided on the parent page, along with a link to the application. The page containing the external image includes identifying information that relates the image to its parent page; for example, if the parent page indicates

"Figure 4 presents the organizational chart," the figure number is included as a cross-reference on the page with the image.

Images are usually provided in Graphics Interchange Format (GIF) or Joint Photographic Experts Group (JPEG) file format. GIF is preferred for graphic images such as drawings, icons, and buttons while JPEG is preferred for photographic images. If the application includes GIF images, they are available in interlaced, rather than non-interlaced, form. An interlaced GIF image becomes progressively clearer as the file is transferred to the browser, while a non-interlaced image is loaded from the top of the picture to the bottom. While both formats take the same amount of time to download, an interlaced one gives more control to users by allowing them to identify the image sooner and determine whether they want to view the entire image or stop it before the transfer is complete. The progressive form of a JPEG image provides a similar advantage compared to the standard form of this image.

The <table> tag can be used to specify a spreadsheet-style arrangement (with cells that contain text or images) or to create a "columns" appearance on a page or in a form. For example, because links can be included in table cells, it is possible to create a composite image by placing a portion of the image in each cell of a table with no borders; when displayed by a browser, the table presents the various images seamlessly and makes the composite behave like an image map (i.e., an image with hyperlinked regions).

The application can emulate dynamic animation by loading a sequence of layered images (using the <meta> tag). With this approach, the animation is handled by the browser itself and does not require a helper application to be downloaded. If this form of animation is implemented, it has beginning and end points and does not play continuously. The image in each frame of the sequence is the same height and width in pixel dimensions and positioned to follow logically after the one that preceded it. An animation sequence consists of no more than ten frames in order to keep downloading time from becoming excessive.

14.2.2 Image Maps

An image map is a graphic image that is divided into multiple regions, each of which navigates to a different destination. If the application creates an image map (e.g., using the <map> tag), it defines specific areas of the image as links and then creates a default link that includes the remainder of the image so that all regions have a defined destination. If a default link is not created, the application provides feedback to users if they select an inactive part of the image map.

Each region of the image map is large enough and has clearly marked boundaries to be easily identified by users. Image maps are interlaced so that users can select a link as soon as they recognize the region they want to select. If an image map is included on a home or navigation page, it is sized so that it can be downloaded within 15 seconds (since these pages are likely to be visited frequently). When an image map is used on a page, the application provides a text-based alternative (using the "alt" attribute) with the same set of links to accommodate users with browsers that cannot support this feature.

One disadvantage of image maps is that they do not provide feedback on the regions of the image that are links. If the application implements client-side image maps, users can obtain this information but only in the status bar of the browser window when the pointer is placed on one of the regions. In addition, compared to text-based navigation pages, image maps provide little information about the content of the destination and are poor information look-up tools as far as speed and detail.

14.2.3 Background Images

A background image is a graphic displayed behind the text on a page. If the application uses a background image, it sets color explicitly and uses a light, neutral color that does not interfere with the readability of foreground text and graphics. Image size is kept to less than 10 kilobytes to minimize download time. As with graphic and photographic images, background images can be provided in GIF or JPEG format. However, interlaced GIF and progressive JPEG formats are not used as background images because the information on the page will not download until the background image is fully interlaced or has progressed to its final resolution.

14.2.4 Animation, Video, and Sound

If the application implements animation or video, the sequence of images has defined beginning and end points, rather than running as an infinite loop. The page from which the animation or video is launched identifies any required helper applications and includes information about the size, projected loading times, and platform requirements involved in accessing and using the video and animation files.

A text marquee is a string of words and phrases that scroll across a page or along the status bar in the browser window. The application does not implement features such as marquees that run constantly or blink because they can be distracting to users. These types of special effects are used only if required to deliver application functionality.

In browsers that support background sounds, a default audio file can be downloaded and played automatically when users view a page; the sound clip can be played once or continuously. The application implements background sounds only if this feature is integral to application functionality. If a background sound is used, the application informs users of its presence and indicates when audio needs to be turned on. The application checks the volume of the sound to ensure that it is not too loud when played. All required helper applications are identified on the page from which the sound can be accessed, and users can select how may times they want the sound to be played.

14.2.5 Helper Applications and Plug-Ins

A helper application is a separate software resource that allows a browser to process or display information (e.g., external media such as graphics, video, animation, and sound) that cannot be handled by the browser itself. If a helper application is used, the page where the resource is launched or triggered indicates which helper application is required and provides links to all

required files. The page also informs users about how the application works, what steps are required to configure the browser to use it, and instructions on how to download it.

A plug-in is software that is integrated into the browser so that multimedia information can be displayed in the browser window without using a separate helper application. If the application makes use of a plug-in, it provides a link to the plug-in that automatically launches the software. The application includes a text explanation with the link so that users who do not have the plug-in are informed rather than getting an error if they select the link. If desired, the application can support downloading standard-format versions of the sound or video media as an alternative to the plug-in. When a plug-in is used, its window is large enough for users to view the contents with minimal scrolling, especially in a horizontal direction. If a page contains multiple text or image files to view using a plug-in, they are kept as small as possible to minimize downloading time.

14.2.6 Frames

Frames define panes in the viewing area of a browser window. Each pane can be viewed, scrolled, and updated independently. Frames can be used in the following ways:

- Frames can freeze the display of standard text in an application. If a set of pages has the same basic header and footer, frames at the top and bottom of the browser window can be used to present this information, with a frame in the middle used to display the unique information on each page (see figure 14-1a). Alternatively, frames can be used when presenting tabular information to lock the row or column titles so they do not scroll out of view.

- Frames can define navigation and information areas for an application. A navigation area at the bottom or side of the browser window (see figure 14-1b) contains the links for navigating within the application. Selecting one of the links displays the page in the information area that occupies the remainder of the window. The set of links in the navigation area is fixed no matter what part of the application users are currently viewing.

- Frames can define individual work areas in an application, with the contents of each area changing based on the links selected by the user. For example, database queries can be submitted in one framed area, and the results viewed in another area (see figure 14-1c).

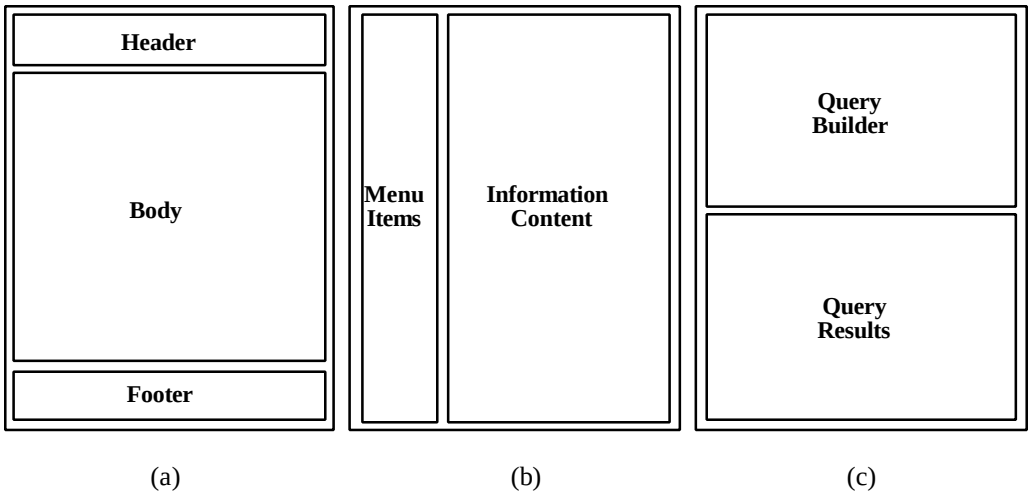

Figure 14-1. Examples of frame placement.

While frames can enhance the overall usability of an application, they can also reduce the available space within the browser window, increase download time, and make it harder to navigate and print within the application. Because browsers differ in the extent to which they support frames, an application that uses frames includes information that it has done so (e.g., using the <noframes> tag) and provides a link that displays application content without frames enabled.

In general, the application uses frames to define no more than three primary viewing areas within the browser window; additional framed areas can be included as needed to create smaller secondary viewing areas (e.g. to display classification markings, row and column headings). The number of frames and their size and placement are based on the type of information being presented, with the same overall layout used throughout the application. An effective layout is one where users are not forced to resize a frame in order to view its contents; if frequent resizing is needed, the frames are not being used effectively and the overall design should be revised. Similarly, an effective layout is one where users do not have to scroll the contents of a frame repeatedly in order to read headings, select from a list of links, or view an inline image.

The application can define frame size in terms of percentage or absolute pixel count or use an asterisk to specify a relative amount. If frame size is defined in terms of pixel count, a standard screen size of 640 x 480 pixels is assumed, with the browser interpolating the meaning of the values for monitors with higher resolutions. The asterisk method is preferred because it allows developers to disregard screen resolution; the asterisked value is automatically computed based on the other values entered. For example, if the first and third values are specified, these frames are always the same size and the second frame is adjusted according to screen resolution.

The application can specify whether a frame can be resized and whether scrollbars are visible on a frame. By default, users can resize the frame, with the browser adding scrollbars only when the text or image is too large to fit within the framed area. The application uses these defaults. The application can also specify (e.g., using the <target> tag) the destination (e.g., the frame) where information is displayed when a link is selected. The default is to display the information in the same frame that contains the link. If the application uses frames as a navigation aid (as described

above), it modifies this default so that the information appears in a frame other than the one with the link (and the contents containing the links remain visible for selection).

14.3 Interactive Capabilities

14.3.1 Forms

A form allows users to enter and interact with the information on a page. The <form> tags specify the interface components that appear on the page, while Common Gateway Interface (CGI) scripting defines the set of commands to be executed when users click on one of the components. The interface components available in a form and their properties are described below.[1] Because browsers differ in how they render <form> tags, the application is tested to ensure that interface components have the desired appearance and behavior on different browsers and platforms.

14.3.1.1 Push Buttons

The <form> tag supports two push buttons: Submit and Reset. The Submit button sends the data on the form to the CGI script for processing, while the Reset button resets the form to its state when initially displayed. There are no properties to specify for these buttons.

The label for the Submit and Reset buttons can be modified as part of their tag (using the "value" attribute) and can be either text or an image. If the label is text, the first letter of each word is capitalized, except for prepositions and articles. If the label is an image, the application provides a text-only version of the button so that users with text-only browsers can identify the action performed by the button. If the application provides default values for any of the components in the form, the label for the Reset button is Reset; if there are no default values, the label for this button is Clear.

14.3.1.2 Radio Buttons

The properties of a radio button are its name and the value that is sent to the CGI script when a button is selected. Because there are multiple radio buttons in a group, each button is assigned a name that uniquely identifies it, and this information is sent to the CGI script to indicate the button selected. The <form> tag (i.e., type = "radio") draws the radio button indicator; the label for each radio button is specified separately.

Radio buttons are presented in groups consisting of at least two but no more than seven options. The label for each option describes the state being set and is placed to the right of the indicator. If a group of radio buttons has a default value, that button is marked as selected (by adding "checked" as part of the tag).

[1] The specifications in this section focus on the appearance of form components on a page; specifications concerning behavior will be provided in a future version of this document.

14.3.1.3 Check Boxes

The properties of a check box are its name and the value that is sent to the CGI script when the check box is selected. The <form> tag (i.e., type = "checkbox") draws the check box indicator; the label for the check box option is specified separately.

Check boxes can be presented individually or in groups. The label for each option describes the state being set and is placed to the right of the indicator. A check box (rather than two radio buttons) is used if the state of a setting can only be on or off. If a check box has a default value, it is marked as selected (by adding "checked" as part of the tag).

14.3.1.4 Text Boxes[2]

The properties of a text box include its name, length, the number of characters that can be entered, and the default text, if any, to be displayed.

The application sets the "size" attribute so that the text box is as long as the longest text string being entered. If the text being entered is a fixed length, the text box is the same length as the text string. The application also sets the "maxsize" attribute so that the text being entered does not automatically scroll out of view when the end of the text box is reached. The application includes a label with the text box that describes what is to be entered and provides cues about format (e.g., unit of measurement) and whether text entry is mandatory or optional. The label is placed to the left of or above the text box and is followed by a colon.

14.3.1.5 Text Areas

The properties of a text area include its name, height and width (i.e., number of rows and columns), and the default text, if any, to be displayed. The application includes a label with the text area that describes the information users are to enter (e.g., "Message text") and indicates if there is a limit on the number of characters that can be entered.

14.3.1.6 List Boxes and Drop-Down List Boxes[3]

The properties of a list are its name, the number of items that are visible, and whether it supports single or multiple selection. A list with one item visible is rendered as a drop-down list box, while a list with multiple items visible is rendered as a scrollable list box. The <form> tag draws the list box, with each item in the list specified separately (i.e., as an <option value>).

The items in the list are presented in sequential order based on the nature of the items and the sequence in which users expect them to occur (e.g., chronological, alphabetical, sequential, functional, by importance). If the list has any default value(s), they are marked as selected (by adding "selected" as part of its tag). The application includes a label with the list box that describes its purpose or contents. The label is placed above or to the left of the list box and is

[2] HTML refers to a text box as a text field.
[3] HTML refers to a list box and a drop-down list box as menus.

followed by a colon. If the label is placed above the list box, it is aligned with the left edge of the box.

14.3.1.7 Arrangement of Components

The application includes all of the components needed to perform a given task on the same page so that users do not have to remember information entered on one page while working on another. If possible, the components are placed so that users can interact with them without scrolling. Components are arranged from left to right, top to bottom on the form in the order in which users are expected to interact with them. A page can contain more than one form, but forms cannot be nested. Related components are placed together and separated from other components by spacing (using <p> and
 tags) and/or separators (using the <hr> tag). If a heading is included with the components, it describes their function or purpose and is either centered or left aligned above the components to which it relates.

Radio buttons and check boxes are arranged in one or more rows or columns. The preferred arrangement is vertical and left-aligned; if placed horizontally, space is sufficient (at least twice the distance between the button/box indicator and its label) so the indicator is paired with the label on the right, not the left. Text boxes are grouped based on sequence of use, frequency of use, or importance. If the labels and text boxes are placed adjacent to each other, the labels and boxes are both left-aligned, or the labels are right-aligned and the boxes are left-aligned, as shown in figure 8-4. If the label is placed above the text box, the label is aligned with the left end of the box. The labels and text boxes can be aligned using the <pre> tag, or they can be placed in the cells of an invisible table in order to justify and align them.

The Submit and Reset push buttons are positioned along the bottom of the form or down the right side of the form, with the Submit button to the left of the Reset button (if the buttons are arranged horizontally) or above the Reset button (if the buttons are placed vertically). If a form contains a single text box, RETURN executes the Submit action; otherwise, users have to click on the Submit button to execute this action. More than one Submit button can be included in a form. A Reset button is included in long forms and is optional in very short forms. If a Reset button is included, it has a label that distinguishes it from the Submit button and is positioned far enough away from this button to minimize the opportunity for users to select it inadvertently.

14.3.2 Java Applets

If the application contains a Java applet, it informs users of its presence and provides information about it, including file size if the applet is large. The application displays a text explanation of the function performed by the applet for users with browsers that understand the <applet> tag but cannot run Java applets. If the applet creates a window that is separate from the browser, it is labeled as such (e.g., “Untrusted Java Applet Window”) and supports window functions in accordance with specifications in sections 4 and 8. If the application uses applets to display interface components such as menus and controls, they are implemented according to specifications in sections 5 and 6 and are arranged according to window design rules in section 8. The application is tested to ensure that it downloads and executes correctly with different browsers and platforms.